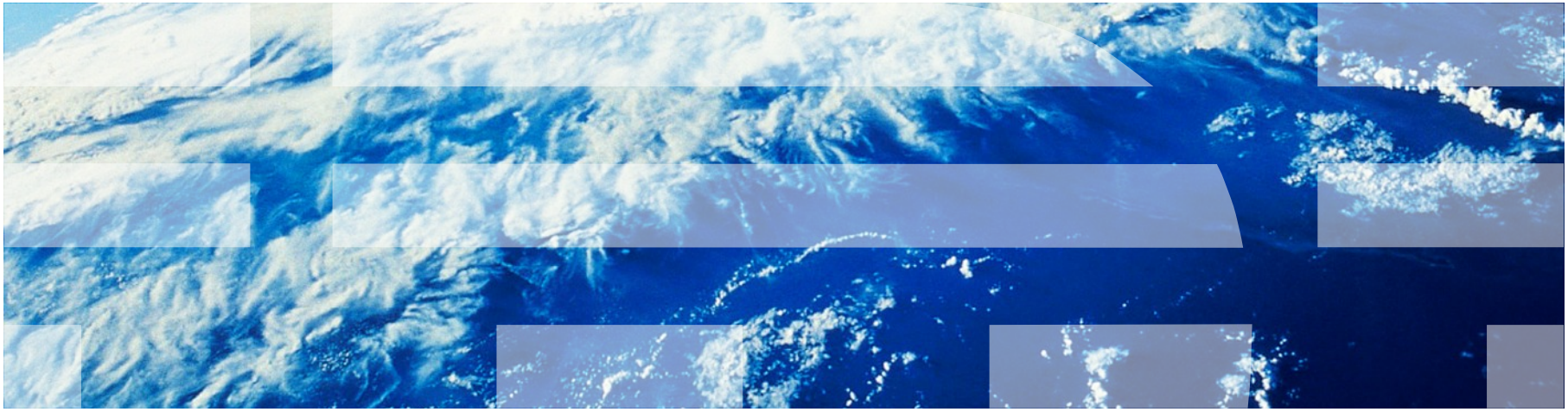
ZVM20: z/VM PAV and HyperPAV Support

Eric Farman, IBM

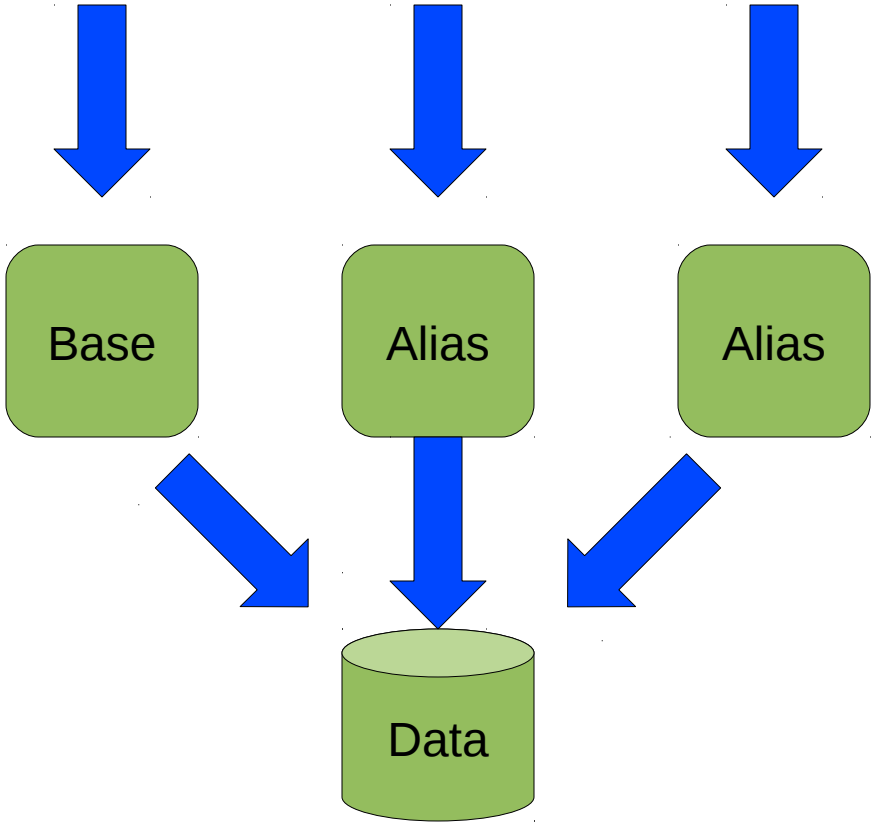


Trademarks

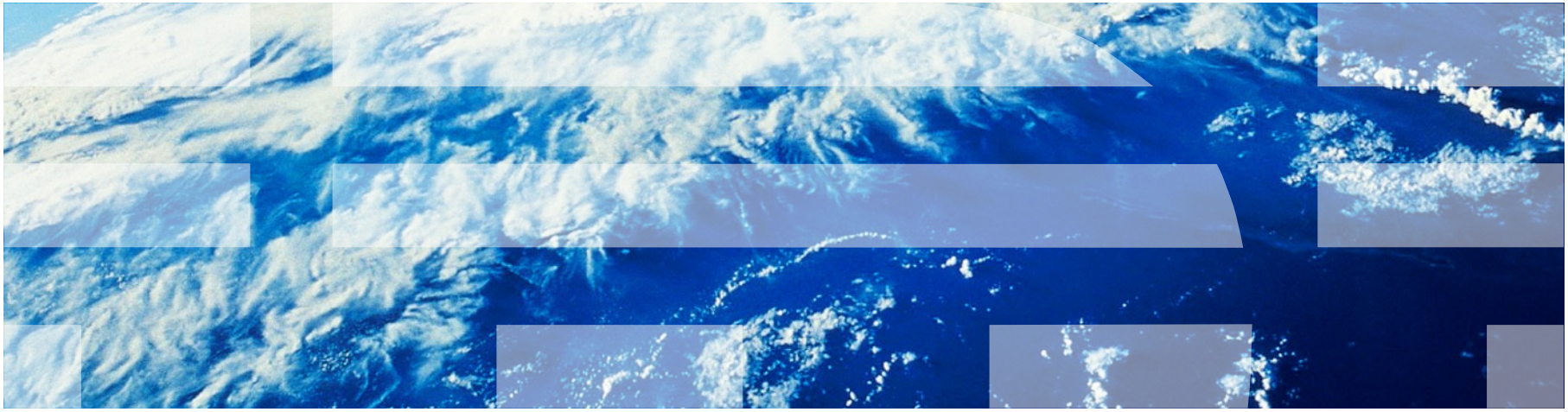
- The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.
  - Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.
  - For a complete list of IBM Trademarks, see **www.ibm.com/legal/copytrade.shtml**
- The following are trademarks or registered trademarks of other companies.
  - Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
  - Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
  - Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
  - Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
  - Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
  - UNIX is a registered trademark of The Open Group in the United States and other countries.
  - Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
  - ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
  - IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.
  - * All other products may be trademarks or registered trademarks of their respective companies.
- Notes:
  - Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
  - IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
  - All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
  - This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
  - All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
  - Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
  - Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- PAV
  - Overview
  - Configuration
  - Commands
  - User Directory
  - Setup Example
  - Dynamic PAV
  - Performance
- HyperPAV
  - Overview
  - Configuration
  - Commands
  - User Directory
  - Configuration File
  - Documentation
- Reference

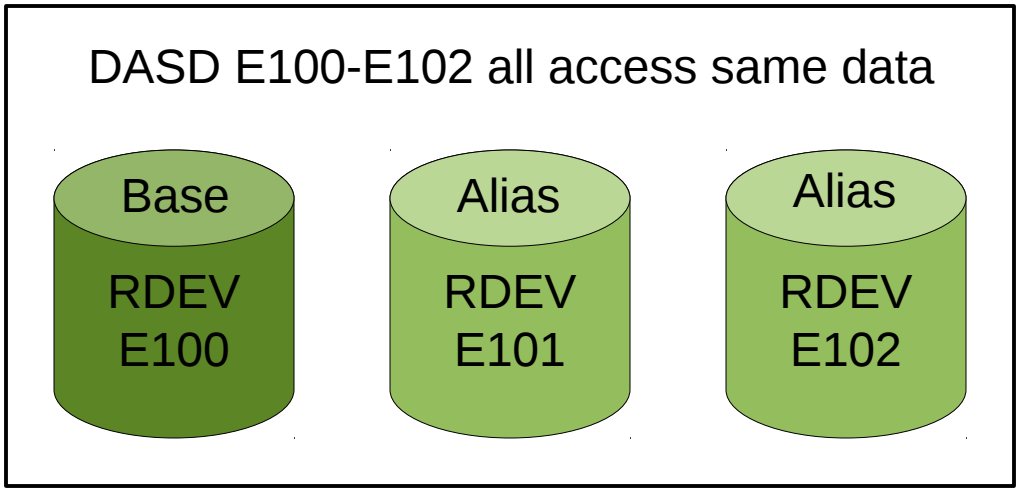


PAV



Overview

- z/VM provides support for the Parallel Access Volumes (PAV) feature of IBM System Storage subsystems.
- With PAV, a real DASD volume is accessed through a Base subchannel (device) and one or more Alias subchannels
  - Volume (represented by Base) shadowed by 1 or more Aliases
  - Looks like multiple separate, real DASD to host operating system



Overview

- z/Architecture allows only 1 active I/O to a single ECKD DASD
- Aliases overcome this restriction providing the ability to have multiple concurrent I/O operations on a DASD
- Allows higher I/O throughput by reducing I/O queuing
- Control unit provides data serialization
- Each I/O request specifies cylinder range:
  - Controller provides shared access for read cylinder ranges
  - Controller provides exclusive access for write cylinder ranges

IOCP Configuration

- PAV volumes can be defined as 3390 Model 2, 3, 9 (inc. mod 27 and 54), or A (EAV) DASD on 3990 Model 3 or 6, 2105, or 2107 Controllers.
- 3380 track-compatibility mode for 3390 Model 2 or 3 DASD are also supported.
- IOCP Statements
  - CNTLUNIT
    - Control units for Bases and Aliases are UNIT=3990, 2105, or 2107
  - IODEVICE
    - Base UNIT=3390 or 3390B or 3380 or 3380B
    - Alias UNIT=3390 or 3390A or 3380 or 3380A

Hardware Console Configuration

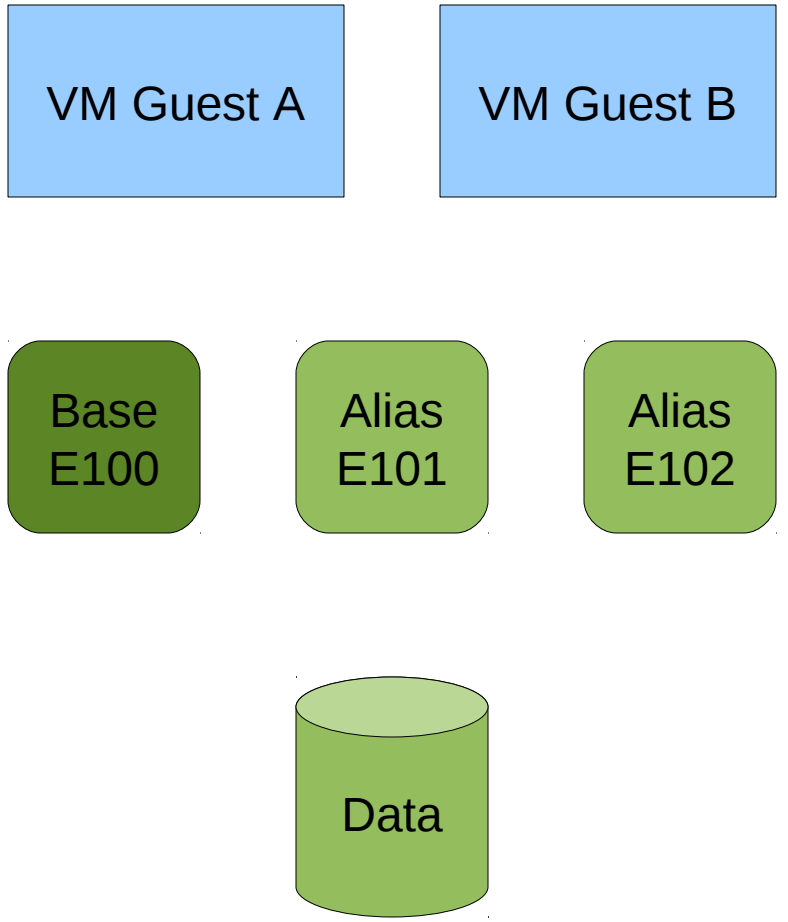
- The assignment of PAV Bases and Aliases is achieved with the control unit's Hardware Management Console (HMC)
- HMC configuration should match IOCP
- Use HMC to initially define which subchannels are Base subchannels, which subchannels are Alias subchannels, and which Alias subchannels are associated with each Base

VM Configuration

- A real Alias subchannel will not come online to VM without an associated real Base subchannel
- A real Base subchannel must have at least 1 associated real Alias subchannel for z/VM to recognize the device as a PAV Base subchannel
- Use the Class B, CP QUERY PAV command to view the current allocation of Base and Alias subchannels

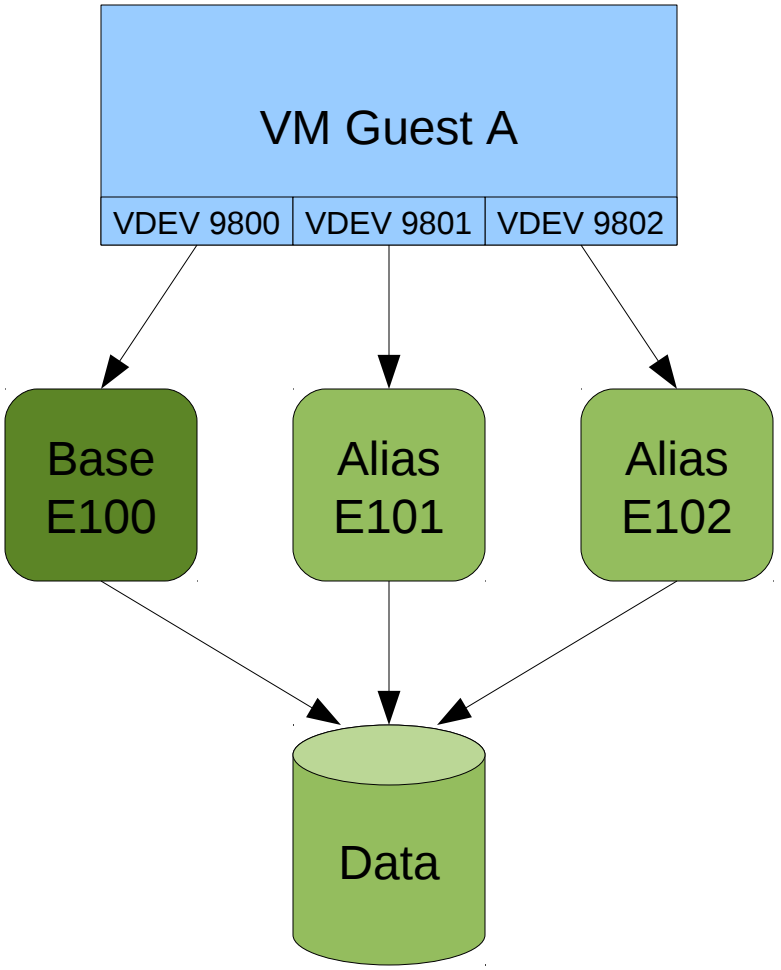
VM Configuration (traditional)

- PAVs were traditionally supported by VM (z/VM 5.2.0 and earlier) for guests as dedicated DASD
- Base and Alias devices could be dedicated to a single guest or distributed across multiple guests
  - The latter is somewhat dangerous!
- Configured to guest(s) with the CP ATTACH command or DEDICATE user directory statement
- Only for guests that exploited the PAV architecture, like z/OS and Linux



VM Configuration Today (Dedicated DASD)

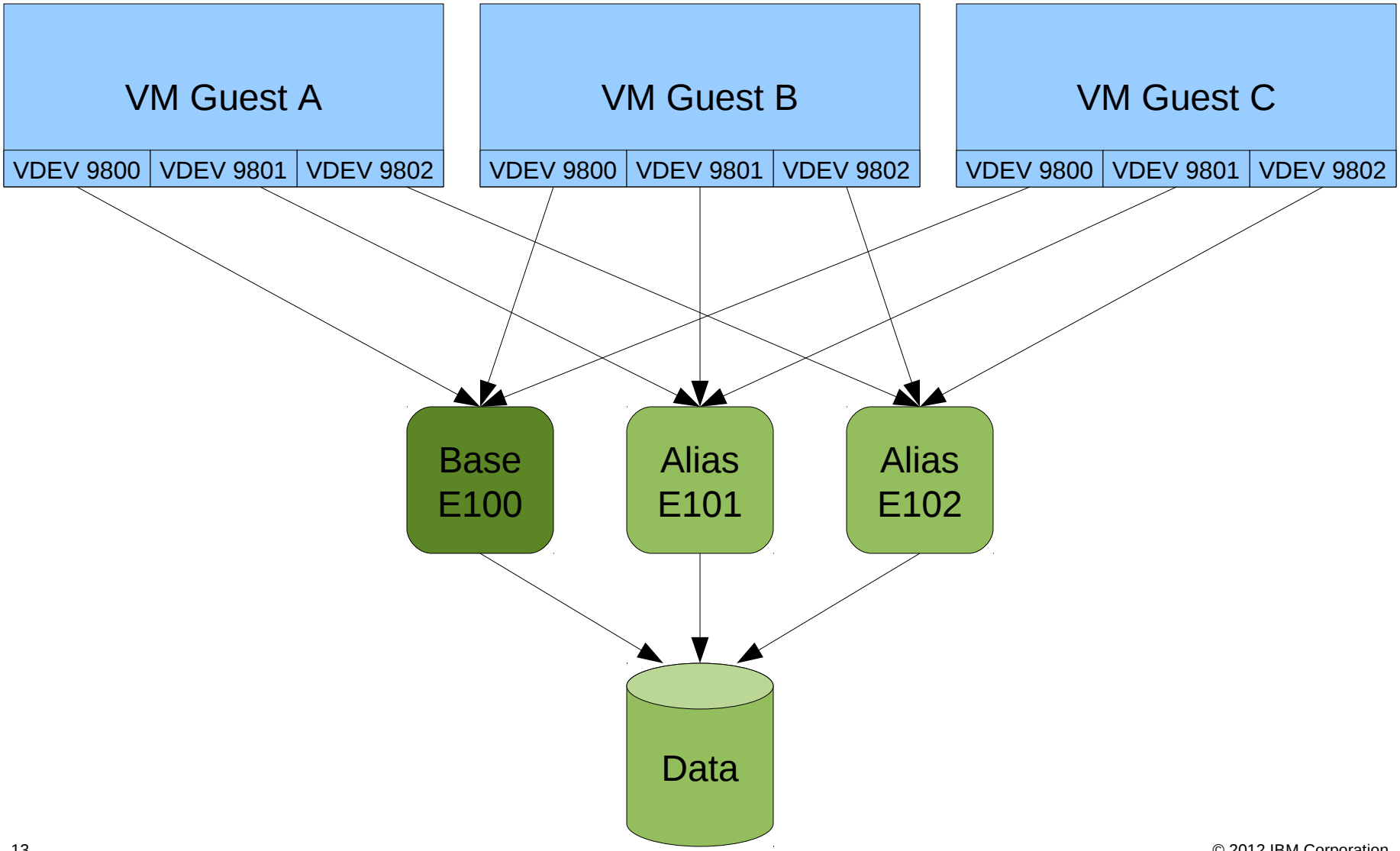
- Today, a Base and its Aliases may only be dedicated to one guest
  - Base must be dedicated first, all alias devices can follow
- Still configured with CP ATTACH command or DEDICATE user directory statement
- For guests that exploit the PAV architecture



VM Configuration Today (Minidisks)

- z/VM provides linkable minidisks for guests that exploit PAV (e.g., z/OS and Linux); see illustration next slide
  - Base minidisks are defined with the existing MDISK or LINK user directory statements (LINK command also supported)
  - Aliases are defined with new PAVALIAS parameter of the DASDOPT and MINIOPT user directory statements or with the new CP DEFINE PAVALIAS command
- z/VM also provides workload balancing for guests that don't exploit PAV (e.g., CMS)
  - Real I/O dispatcher queues minidisk I/O across system attached Aliases
  - Minidisks are defined as in the past; nothing has changed.

VM Configuration Today (Minidisks)



ATTACH Command

- When attaching PAV DASD to a guest, the Base must be attached first before any associated Alias. An associated Alias can only be attached to the same guest as the Base.
- When attaching PAV DASD to the system, the Base must be attached first before any associated Alias.
  - Aliases can be attached to the system and are exploited for VM I/O if they contain temporary disk (TDSK) or minidisk (PERM) allocations.
  - Other CP volume allocations (e.g., PAGE) receive no benefit from system attached Aliases.

DETACH Command

- When detaching PAV DASD from a guest, all dedicated Aliases associated with a particular Base must be detached from the guest before the Base can be detached.
- When detaching PAV DASD from the system, all system attached Aliases associated with a particular Base must be detached from the system before the Base can be detached.

Minidisk Cache (MDC)

- Minidisk cache settings apply to the Base and are inherited by its Aliases
- SET MDCACHE command may not be used with Aliases; results in error

DEFINE PAVALIAS Command

Privilege Class G

```
>>--DEFine--PAValias--vdev--.------.--BASE--basevdev-------------><
                                '-FOR-'
```

- The DEFINE PAVALIAS command is used to create new virtual PAV Alias minidisks. Function can also be accomplished by using the DASDOPT and MINIOPT user directory statements.
- Newly defined virtual Alias is automatically assigned to a unique underlying real PAV Alias.
- The command will fail if no more unique real Aliases are available to be associated with the virtual Alias (per guest virtual machine).

Query Virtual PAV

Privilege Class G

```
           .-Virtual----.       .-ALL---------------.
>>--Query--'------------'--PAV--+-------------------+-----------><
                                | <---------------< |
                                '--.-vdev--------.--'
                                   '-vdev1-vdev2-'
```

Dedicated Responses

```
QUERY VIRTUAL PAV ALL
PAV BASE  0290 ON E100 WIL3
PAV ALIAS 0291 ON E101 WIL3 FOR BASE 0290
```

Minidisk Responses

```
QUERY VIRTUAL PAV ALL
PAV BASE  0290 ON E100 WIL3 ASSIGNED E100
PAV ALIAS 0291 ON E101 WIL3 ASSIGNED E101 FOR BASE 0290
```

User Directory

- DASDOPT
  - Used for Full-Pack Minidisks
    - MDISK vdev devtype DEVNO rdev mode

      DASDOPT PAVALIAS vdev
    - MDISK vdev devtype 0 END volser mode

      DASDOPT PAVALIAS vdev-vdev
    - LINK userid vdev1 vdev2 mode DASDOPT PAVALIAS vdev.numDevs
- MINIOPT
  - Used for Non-Full-Pack Minidisks
    - MDISK vdev devtype 100 50 volser mode

      MINIOPT PAVALIAS vdev
    - LINK userid vdev1 vdev2 mode

      MINIOPT PAVALIAS vdev-vdev

- PAVALIAS option of DASDOPT and MINIOPT statements are used to create virtual PAV Alias minidisks for a guest.
- DASDOPT and MINIOPT should follow the MDISK or LINK statement associated with the virtual Base.
- DASDOPT and MINIOPT may be continued on multiple lines with trailing commas.
- Can have more Aliases in user directory than exist in hardware. Virtual Aliases will be assigned in ascending order until the real associated Aliases are exhausted. This will not prevent logon!
- Use DEDICATE vdev rdev for all dedicated PAV Base and Alias devices.

Setup example for Linux exploiting PAV minidisks

```
Base device predefined in user directory:
MDISK 200 3390 DEVNO E100 WR

q pav
Device E100 is a base Parallel Access Volume with the following aliases: E101
Device E101 is an alias Parallel Access Volume device whose base device is E100

attach E100 to system
DASD E100 ATTACHED TO SYSTEM WIL6 PAV BASE
attach E101 to system
DASD E101 ATTACHED TO SYSTEM WIL6 PAV ALIAS

define pavalias 201 for base 200
DASD 201 DEFINED

query virtual pav all
PAV BASE  0200 ON E100 WIL6 ASSIGNED E100
PAV ALIAS 0201 ON E101 WIL6 ASSIGNED E101 FOR BASE 0200
```

Configure Linux LVM to use virtual PAV Base 200 and Alias 201 as a single logical volume.
For details, see Linux “How to Improve Performance with PAV” whitepaper at:
http://www-128.ibm.com/developerworks/linux/linux390/june2003_documentation.html

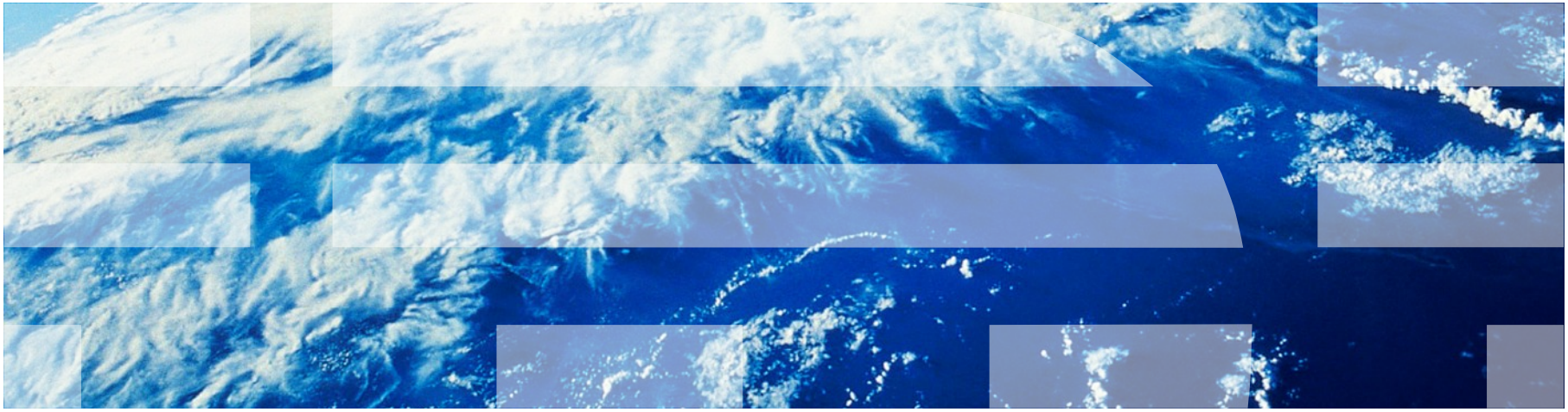
Dynamic PAV

- Dynamic PAV is the ability to re-associate an Alias device from one Base to another
- Guest issued dynamic PAV operation to a dedicated Alias:
  - Real (and virtual) Alias to Base association will change as long as the new Base is dedicated to the same guest. Otherwise, the dynamic PAV operation fails.
- Guest issued dynamic PAV operation to an Alias minidisk:
  - Full-pack minidisks only. Otherwise, the Dynamic PAV operation fails.
  - There must be a unique real Alias available in which to associate the virtual Alias (per guest machine). Otherwise, the Dynamic PAV operation fails.
  - Only the virtual configuration is altered. The real Alias to Base association never changes for minidisks.
- Out-board (control unit) initiated dynamic PAV operations:
  - All Alias minidisks associated with a real system attached Alias will be detached from their guests.
  - A dedicated Alias will behave as if guest issued the dynamic PAV operation.

Performance

- Dedicated DASD
  - Performance metrics for dedicated DASD are solely up to guest virtual machine; VM issues real I/O as indicated by guest.
- Minidisks
  - Useful for environments where I/O queuing occurs (see Performance Toolkit FCX168 report, or equivalent)
  - Performance gains may be realized only when minidisks are shared among guests with multiple LINK statements or when multiple minidisks reside on a real volume
  - Performance gains are achieved by multiplexing the I/O operations requested on each guest minidisk over the appropriate real PAV Base and Alias subchannels
  - Performance varies depending on controller model and read-write mix
  - “Law of Diminishing Returns”; defining more Aliases than needed can lower performance
  - Success Criterion: Response time equals service time (no wait queue)
  - For details see,
    - http://www.vm.ibm.com/perf/pavmdc.html
    - http://www.vm.ibm.com/perf/reports/zvm/html/520pav.html

HyperPAV

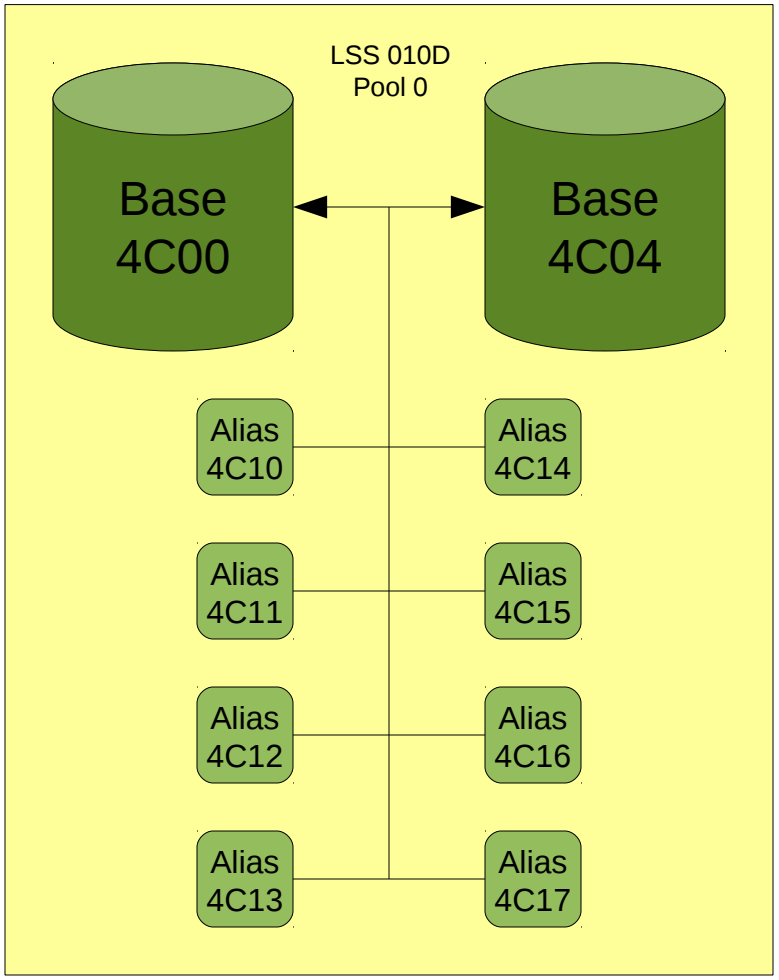


Overview

- New feature of the DS8000 (only) that removes the static Alias to Base binding associated with traditional PAVs
- Alias and Base volumes are pooled per each LSS. An Alias can be associated with any Base in the Pool; done by host on each I/O request.
- Makes traditional Dynamic PAV obsolete
- VM support for dedicated DASD and Fullpack Minidisks available in z/VM 5.4.0 and later

Overview

- VM dedicated DASD support via CP ATTACH command or DEDICATE user directory statement
- VM Minidisk Support:
  - workload balancing for guests that don't exploit HyperPAV
  - linkable full-pack minidisks for guests that do exploit HyperPAV
  - New CP DEFINE HYPERPAVALIAS command creates HyperPAV Alias minidisks for exploiting guests
  - Current exploiters of HyperPAV are z/VM, z/OS and Linux (SLES11, RHEL6)
  - Restricted to fullpack minidisks for exploiting guests



Configuration

- HyperPAV Base and Alias subchannels are defined on control unit's Hardware Management Console and in IOCP no differently than traditional PAVs
- HyperPAV hardware, priced feature enables floating Alias function associated with the HyperPAV architecture for each LSS (logical control unit)
- Operating system host determines which LSS (logical control unit) is in HyperPAV vs. traditional PAV mode

Configuration

- A real HyperPAV Alias subchannel will not come online unless a HyperPAV Base exists in the same hardware Pool.
- A real HyperPAV Base subchannel needs at least 1 HyperPAV Alias in the same hardware Pool for z/VM to recognize the device as a HyperPAV Base subchannel.
- Use the Class B, CP QUERY PAV command to view the current HyperPAV Base and Alias subchannels along with their associated Pools.

ATTACH / DETACH Commands

- Unlike traditional PAV DASD, HyperPAV Base and Alias devices can be attached and detached to/from a guest or the system in any order. There is no Base before Alias (or vise-versa) restrictions.
- HyperPAV Aliases can be attached to the system and are exploited for VM I/O if they contain temporary disk (TDSK) or minidisk (PERM) allocations.
- Other CP volume allocations receive no benefit from system attached HyperPAV Aliases.

Minidisk Cache (MDC)

- Minidisk cache settings do not apply to HyperPAV Aliases. Cache setting are only applicable to HyperPAV Base devices.
- SET MDCACHE command may not be used with HyperPAV Aliases; results in error

DEFINE HYPERPAVALIAS Command

Privilege Class G

```
>>--DEFine--HYPERPAValias--vdev--.------.--BASE--basevdev---------><
                                    '-FOR-'
```

- The DEFINE HYPERPAVALIAS command is used to create new virtual HyperPAV Alias minidisks.
- A newly defined virtual Alias is automatically assigned to a unique underlying real HyperPAV Alias (in the same real hardware Pool as the Base).
- The command will fail if no more unique, real Aliases are available in the real hardware Pool to be associated with the virtual Alias (per guest virtual machine).
- There can only be 254 Aliases per Pool; and a limit of 16,000 Pools per LPAR.
- Command currently restricted to Full-Pack minidisks.

Query Virtual PAV Command

Dedicated

```
QUERY VIRTUAL PAV ALL
HYPERPAV BASE  0200 ON E100 YAC001 POOL 1
HYPERPAV ALIAS 0201 ON E101 POOL 1
```

Minidisks

```
QUERY VIRTUAL PAV ALL
HYPERPAV BASE  0200 ON E100 YAC001 ASSIGNED E100 POOL 1
HYPERPAV ALIAS 0201 ASSIGNED E101 POOL 1
```

SET CU Command

- The SET CU command is used to set the Parallel Access Volume function level of each applicable control unit (specified via controller's ssid).
- Default is either HYPERPAV_allowed or PAV_allowed depending on the installed capabilities of each control unit.
- HYPERPAV_allowed can't be set if capability is not available on the control unit.
- All Alias devices in the specified control unit (ssid) must be off-line when changing from or to the HYPERPAV_allowed setting.
- Command applies to only first-level VM images; error occurs otherwise.
- New QUERY CU command displays the PAV and HYPERPAV capabilities of applicable DASD control units.

```
              .-DASD-.                              <-------------<
>>--SET--CU--'------'--.-HYPERPAV_allowed-.---.-ssid------.-----><
                       |-PAV_allowed------|   '-ssid-ssid-'
                       '-NOPAV_allowed----'
```

User Directory

- Use the COMMAND user directory statement with the DEFINE HYPERPAVALIAS command to create virtual HyperPAV Alias minidisks
- COMMAND statements must appear before all device definition statements, like MDISK and LINK statements for the Base minidisks
- Examples:
  - COMMAND DEFINE HYPERPAVALIAS vdev FOR BASE basevdev
  - MDISK basevdev devtype DEVNO rdev mode
  - COMMAND DEFINE HYPERPAVALIAS vdev FOR BASE basevdev
  - MDISK basevdev devtype 0 END volser mode
  - COMMAND DEFINE HYPERPAVALIAS vdev FOR BASE basevdev
  - LINK userid sourcevdev basevdev mode
- Use DEDICATE vdev rdev for all dedicated HyperPAV Base and Alias devices

Configuration File

The following new system configuration file statements are useful for managing HyperPAV devices:

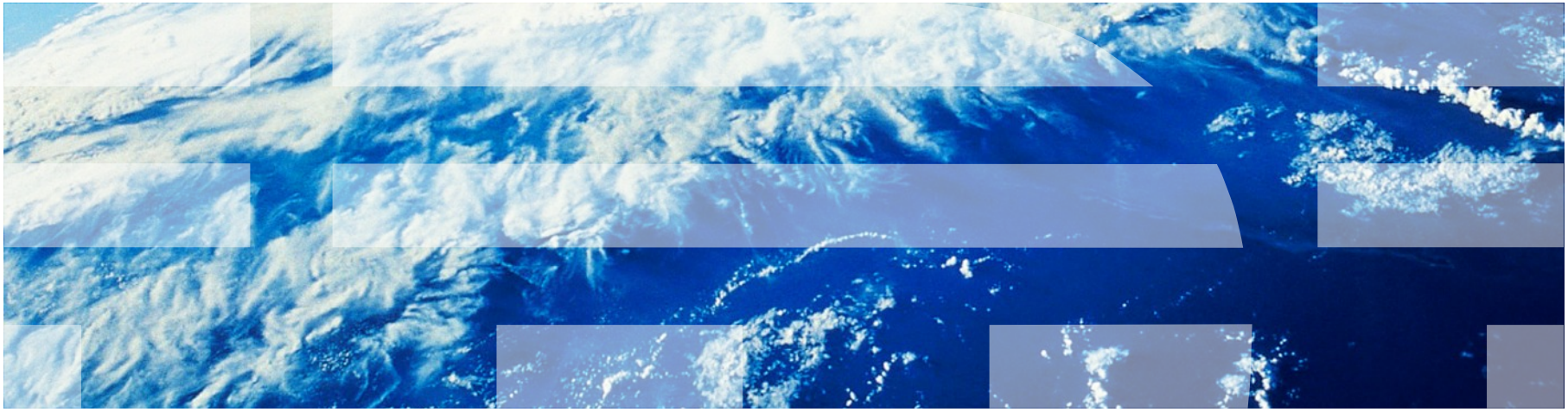
- SYSTEM_Alias - Specifies HyperPAV Alias devices to be attached to the system at VM initialization.
- CU - Defines how VM initializes specific control units. Similar to the CP SET CU command (i.e., sets controller PAV mode).

Documentation

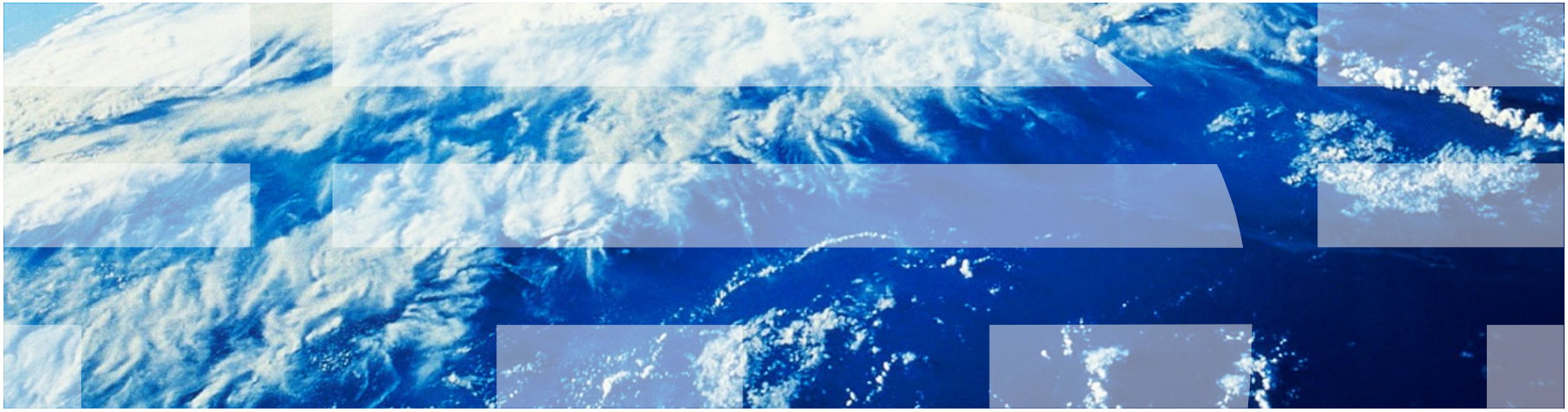
- CP Command Reference
  - Command details
- CP Planning and Administration Guide
  - User Directory Statements
  - Configuration File Statements
  - “DASD Sharing Chapter” with “Using IBM Parallel Access Volumes” section
- CP Messages and Codes
  - New and changed messages
- Web
  - http://www.vm.ibm.com/storman/pav/
  - http://www.vm.ibm.com/perf/pavmdc.html
  - http://www.vm.ibm.com/perf/reports/zvm/html/520pav.html
  - http://www-128.ibm.com/developerworks/linux/linux390/june2003_documentation.html

Fin

Eric Farman
farman@us.ibm.com
z/VM I/O Development



Reference



PAV with Non-Fullpack Minidisks

- Neither z/VM, nor its guest operating systems, could uniquely identify distinct non-fullpack minidisks
  - Thus, I/O may inadvertently be driven down an incorrect alias
- z/VM 5.4 and later correctly address this problem
  - PTFs available for earlier releases via APAR VM64273
- Co-requisite fixes are required for guest operating systems
  - z/OS APARs OA22161 and OA25151 are available for z/OS v1.7 through v1.9
  - Linux on System z: see problem ID 34345 in patch 21 of the October 2005 stream on DeveloperWorks
    - http://www.ibm.com/developerworks/linux/linux390/linux-2.6.16-s390-21-october2005.html
    - Included in modern distro's from SUSE and Red Hat
- Patches can be applied in stages, but all need to be present in order to correct this problem.

Restrictions

- A traditional PAV, real Alias may be attached to a guest or SYSTEM only after its associated real Base has been attached to the same guest or SYSTEM.
- A tradition PAV, real Base may be detached from a guest or SYSTEM only if all of its associated Aliases are already free.
- A traditional PAV, real Alias will not come online to VM without an associated real Base. Also, a real Base must have at least one associated real Alias for VM (for example, the QUERY PAV command) to recognize the device as a PAV.
- A real Base cannot be changed or deleted with the SET RDEVICE, DELETE RDEVICE, DELETE DEVICE, or MODIFY DEVICE commands unless all associated real Aliases have been deleted with the DELETE RDEVICE command.

Restrictions

- CMS does not support virtual Aliases (whether traditional PAV or HyperPAV). Defining these virtual devices under CMS can cause similar damage that can be caused by issuing multi-write (MW) links.
- A virtual Alias (whether traditional PAV or HyperPAV) cannot be IPLed.
- PAV Aliases (whether traditional PAV or HyperPAV) can not be used as VM installation volumes (for example, do not use for the SYSRES volume).
- VM Paging and Spooling operations do not take advantage of PAVs (traditional or HyperPAV). It is recommended that PAGE and SPOOL areas be placed on DASD devices dedicated to this purpose.
- Virtual HyperPAV devices can only be defined as Full-Pack minidisks.
- Diagnoses x18, x20, xA4, x250, and the *BLOCKIO system service do not support HyperPAV Alias devices since there is no means for specification of the associated Base. An attempt to do so will result in an error.